REMARKS

Status of the Claims

- Claims 1-16 are pending in the Application after entry of this amendment.
- Claims 1-16 are rejected by the Examiner.
- Claims 1, 3, 10, and 11 are amended by Applicant.

Claim Rejections Pursuant to 35 U.S.C. §103

Claims 1-16 stand rejected pursuant to 35 U.S.C. §103(a) as being anticipated by U.S. Patent No. 5,873,080 to Coden et al. (Coden) in view of U.S. Patent No. 6,697,799 to Neal et al. (Neal). The Applicant respectfully traverses the rejection.

Coden teaches a system and method that uses a single query with one or more media types to search a collection of multimedia documents in a database and produce a single combined result. (See col. 3, lines 33-36.) The invention of Coden has a combined query section which allows a user to input a single query with more than one media type. The combined query has a query data structure which is submitted to a query interface. The query interface stores the different parts of the query and then parses the query to separate the query according to type. A query object is built for each query type. The query interface translates each of the query objects by query type into queries which are understood by the application programming interface that is designed for a particular search engine. The query interface then distributes the queries to the appropriate search engines. (See col. 3 lines 40-51).

Referring to Figure 1 of Coden, The User Interface (UI, 125) collects user inputs and passes the user inputs to the Enhanced User Interface Support (EUIS, 120) program. (See col. 5 lines 46-55). The EUIS also parses the query into query objects each having a single media type and each being suitable for a particular search engine (162, 164, 166). (See col. 5, lines 55-57). The output of the EUIS is a combined query object (110). The combined query object comprises of one or more query objects (112-117) and one or more query operator objects (118) (See col. 6 lines 1-3). Each query object defines a query condition to be satisfied by a set of one or more parts (153) and/or one or more documents (151) of the database (150) that will be returned as part of a collection of result objects (182, 184, and 186) of each of the respective search engines (162, 164, and 166) (See col. 6, lines 22-27).

In Coden, when the user executes the search, the EUIS passes the combined query to the query interface (130). The query interface analyzes each query object to determine the media type of the query object and the search engine(s) suited to execute the query object. (See col. 6 lines 32-36).

Thus, Applicant understands that, in Coden, a User Interface (125) accepts user inputs until a query may be formed. The user query is then passed to the EUIS (120). The EUIS then parses the query into query objects according to media type. The EUIS thus produces a combined query (110). Each query object of the combined query is already sorted for a media type and specifies a query condition that can be satisfied by a respective search engine. Upon execution of the search, the combined query having the pre-sorted query objects is passed to the query interface (130). The query interface examines the pre-sorted query objects and determines which search engine is to be used for each query object.

Applicant respectfully submits that Coden is a valuable teaching, but the structure and operation of Coden is different than the structural and operational elements recited in amended Claim 1. An example of some of the claimed structure and functionality may be found in the non-limiting example of Figure 3 of the as-filed application which provides support for the amendment of Claim 1.

Claim 1 recites a method of distributing portions of a query over two or more execution engines, the method includes receiving an input query into a first analysis engine of serially cascaded analysis engines. The first of the serially cascaded analysis engines identifies a portion of the input query that can be processed by a first execution engine. That identified portion is compiled forming a first compiled portion.

The original input query is then rewritten to form a first rewritten query. In the first rewritten query, the identified portion of the input query is removed and replaced by a first placeholder. The rewritten first query, which includes the first placeholder, is then input in its entirety, into the succeeding analysis engine in the serial cascade of analysis engines. The rewritten query including the first placeholder is thus provided as an input from an analysis engine to the succeeding serially connected analysis engine.

For example, in Figure 3 of the as-filed application, analysis engines A though C (310, 320, and 330) are serially cascaded. An input query 300 is received by engine A (310) and a first portion (on path 310) of the query can be compiled. The rewritten query (on path

314), including a first placeholder, is input to the next succeeding analysis engine B (320). Thus, as recited in Claim 1, the entire portion of the first rewritten query including the first placeholder is passed to a second analysis engine of the serially cascaded analysis engines.

The second analysis engine can then identify a portion of the first rewritten query that can be processed by a second execution engine. Then, the identified portion of the first rewritten query is compiled (see path 322 in Figure 3). The result of the serially cascaded analysis engines is that the input query is distributed over the first execution engine and the second execution engine for sequential execution.

Applicant finds that the teaching of Coden fails to teach serially cascaded analysis engines that identify a portion of an input query that can be processed by an analysis engine and then rewrite the input query to form a rewritten query such that the identified portion of the input query is removed and replaced by a placeholder and further, where the entire portion of the rewritten query, including the placeholder, is passed to a to a second analysis engine of the serially cascaded analysis engines.

Applicant submits that Coden does not disclose serially cascaded analysis engines as described above and recited in Claim 1 because Coden teaches that the query interface (130) analyzes query objects, already sorted by object type, to determine which search engine should receive the query object. Although Coden teaches in col. 6 lines 55 that the query interface (130) translates each query object into a series of API calls to their respective search engines and that the query interface (130) provides any formatting necessary to make the query objects compatible with the API, this does not teach the a rewritten query, including a placeholder is passed to a subsequent analysis engine in a serially connected cascade of analysis engines as is recited in amended Claim 1.

Applicant notes that the present Office Action, dated 12/7/06, states on page 5 that Coden does not disclose receiving an input query into a cascaded analysis engine. Applicant agrees. However, Applicant disagrees that Neal teaches the elements not taught by Coden. Specifically, Applicant respectfully submits that Neal, like Coden, fails to teach serially cascaded analysis engines that identify a portion of an input query that can be processed by an analysis engine and that rewrite the input query to form a rewritten query such that the identified portion of the input query is removed and replaced by a placeholder and further,

where the entire portion of the rewritten query, including the placeholder, is passed to a to a second analysis engine of the serially cascaded analysis engines.

Neal teaches an automatic item classifier that uses the item's attributes based on a classification schema and a knowledge base. The invention of Neal can include selecting a first attribute of the item, designating a first search strategy comprising the value of the first attribute applied to operate upon data records in a first database, selecting a second attribute of the item; designating a second search strategy comprising the value of the second attribute applied to operate upon a second subset of data records in a second database, forming a search hierarchy comprised of the first search strategy followed by the second search strategy, executing the search hierarchy, and selecting at least one classification for the item based on the search. (See Neal, Abstract)

Applicant notes that Figure 7 of Neal is described as a flow diagram of a cascaded search process across multiple databases (see col. 2 lines 10-11). Neal discloses that steps 706, 710, and 714 represent different search strategies (col. 18, lines 5-7). Also, if one search strategy is successful, then the process of Figure 7 may stop (col. 18, lines 27-28). Alternately, an exhaustive search may be performed by using all of the strategies (col. 18, lines 30-33). However Applicant finds that there is no teaching in Neal that corresponds to serially cascaded analysis engines that identify a portion of an input query that can be processed by an analysis engine that rewrites the input query to form a rewritten query such that the identified portion of the input query is removed and replaced by a placeholder and further, where the entire portion of the rewritten query, including the placeholder, is passed to a to a second analysis engine of the serially cascaded analysis engines as recited in amended Claim 1.

Applicant has amended independent Claims 10 and 11 similarly to that of Claim 1. Since Coden fails to teach elements of amended Claim 1 and since Neal also fails to teach some of those same elements, then the combination of Coden and Neal also fail to teach all of the elements in the pending amended claims. Specifically, the combination of Coden and Neal fails to comprehensively teach serially cascaded analysis engines that identify a portion of an input query that can be processed by an analysis engine that rewrite the input query to form a rewritten query such that the identified portion of the input query is removed and replaced by a placeholder and further, where the entire portion of the rewritten query,

including the placeholder, is passed to a to a second analysis engine of the serially cascaded analysis engines as recited in the pending independent claims.

Since all elements of the amended independent claims are not taught by the combination of Coden and Neal, then the combination does not establish a prima facie case of obviousness under 35 U.S.C. §103(a) per MPEP §2143.03. Applicant respectfully requests withdrawal of the 35 U.S.C. §103(a) rejection of Claims 1-16 because the claims patentably define over the cited art.

Applicant amends Claim 3 to more clearly recite that which the inventor regards as the invention. Applicant finds support for this amendment in paragraph 0042 of the present application.

Conclusion

Applicant respectfully submits that all pending claims patentably define over the cited art. Applicants respectfully request reconsideration and withdrawal of the rejections. A Notice of Allowance for all pending claims is requested.

Respectfully submitted,

Date: February 6, 2007

/Jerome G. Schaefer/

Jerome G. Schaefer
Registration No. 50,800

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439